

```
index-1 (1).py  Welcome  JS oop.js 5  JS functional.js  JS procedural.js 1  # app.css  index.html X
C: > Users > Wamwe > Downloads > index.html > html > head > script
1  <!DOCTYPE html>
2  <html lang="en">
3      <head>
4          <meta charset="UTF-8">
5          <meta name="viewport" content="width=device-width, initial scale=1.0">
6          <meta http-equiv="X-UA-Compatible" content="ie=edge" />
7          <title>Programming Paradigms</title>
8          <link rel="stylesheet" href="app.css" />
9          <script src="functional.js" defer></script>
10     </head>
11     <body>
12         <form id="user-input">
13             <div class="form-control">
14                 <label for="username">Username</label>
15                 <input type="text" id="username" />
16             </div>
17             <div class="form-control">
18                 <label for="password">Password</label>
19                 <input type="password" id="password" />
20             </div>
21             <button type="submit" style="background-color: blue;color: white;">Create User</button>
22         </form>
23     </body>
24 </html>
```

```
index-1 (1).py  Welcome  JS oop.js 5  JS functional.js X  JS procedural.js 1  # app.css  index.html
C: > Users > Wamwe > Downloads > JS functional.js > signupHandler
1  const REQUIRED = 'REQUIRED';
2  const MIN_LENGTH = 'MIN_LENGTH'
3
4  function validate(value, flag, validatorValue) {
5      if (flag === REQUIRED) {
6          return value.trim().length > 0;
7      }
8      if (flag === MIN_LENGTH) {
9          return value.trim().length > validatorValue;
10     }
11 }
12
13 function getUserInput(inputElementId) {
14     return document.getElementById(inputElementId).value;
15 }
16
17 function signupHandler(event) {
18     event.preventDefault();
19
20     const enteredUsername = getUserInput('username');
21     const enteredPassword = getUserInput('password');
22
23
24 }
```

Username

mawkamau@usiu.ac.ke

Password

.....

Create User

```
File Edit Selection View Go Run ... Search
index-1 (1).py Welcome JS oop.js 5 JS functional.js JS procedural.js 1 # app.css index.html X
C: > Users > Wamwe > Downloads > index.html > html > head > script
1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <meta charset="UTF-8">
5     <meta name="viewport" content="width=device-width, initial scale=1.0">
6     <meta http-equiv="X-UA-Compatible" content="ie=edge" />
7     <title>Programming Paradigms</title>
8     <link rel="stylesheet" href="app.css" />
9     <script src="procedural.js" defer></script>
10  </head>
11  <body>
12    <form id="user-input">
13      <div class="form-control">
14        <label for="username">Username</label>
15        <input type="text" id="username" />
16      </div>
17      <div class="form-control">
18        <label for="password">Password</label>
19        <input type="password" id="password" />
20      </div>
21      <button type="submit" style="background-color: blue;color: white;">Create User</button>
22    </form>
23  </body>
24 </html>
```

```
index-1 (1).py Welcome JS oop.js 5 JS functional.js JS procedural.js 1 X # app.css <> index.html
C: > Users > Wamwe > Downloads > JS procedural.js > signupHandler
1 const form=document.getElementById('user-input');
2
3 function signupHandler(event) {
4     event.preventDefault();
5
6     const userNameInput = document.getElementById('username');
7     const enteredUsername = userNameInput.value;
8
9     const passwordInput = document.getElementById('password');
10    const enteredPassword = passwordInput.value;
11
12    if (enteredUsername.trim().length===0) {
13        alert('Invalid input - username must not be empty!');
14        return;
15    }
16    if (enteredPassword.trim().length <= 5){
17        alert('Invalid input - password must be six characters or longer.')
18        return;
19    }
20
21    const user = {
22        userName: enteredUsername
23        password: enteredPassword
24    };
25
26    console.log(user);
27    console.log('Hi, I am ' + user.userName);
28 }

```

```
File Edit Selection View Go Run ... Search
index-1 (1).py Welcome JS oop.js 5 JS functional.js JS procedural.js 1 X # app.css <> index.html
C: > Users > Wamwe > Downloads > JS procedural.js > signupHandler
3 function signupHandler(event) {
18     return;
19 }
20
21 const user = {
22     userName: enteredUsername
23     password: enteredPassword
24 };
25
26 console.log(user);
27 console.log('Hi, I am ' + user.userName);
28 }
29
30 form.addEventListener('submit', signupHandler);

```

Username

mawkamau@usiu.ac.ke

Password

.....

Create User

The image shows a VS Code editor with two files open. The top file is `index.html`, which contains the HTML structure for a user creation form. The bottom file is `oop.js`, which contains a JavaScript class `User` with a constructor and a `greet` method.

```
1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <meta charset="UTF-8">
5     <meta name="viewport" content="width=device-width, initial scale=1.0">
6     <meta http-equiv="X-UA-Compatible" content="ie=edge" />
7     <title>Programming Paradigms</title>
8     <link rel="stylesheet" href="app.css" />
9     <script src="oop.js" defer></script>
10  </head>
11  <body>
12    <form id="user-input">
13      <div class="form-control">
14        <label for="username">Username</label>
15        <input type="text" id="username" />
16      </div>
17      <div class="form-control">
18        <label for="password">Password</label>
19        <input type="password" id="password" />
20      </div>
21      <button type="submit" style="background-color: blue;color: white;">Create User</button>
22    </form>
23  </body>
24 </html>
```

```
1 class User {
2   constructor(uName, uPassword) {
3     this.userName = uName;
4     this.password = uPassword
5   }
6
7   greet( {
8     console.log('Hi, I am ' + this.userName);
9   })
10 }
```

Username

mawkamau@usiu.ac.ke

Password

.....

Create User

```
index-1 (1).py Welcome JS oop.js 5 JS functional.js JS procedural.js 1 # app.css x index.html ...
C: > Users > Wamwe > Downloads > # app.css > label
1  html{
2    font-family: sans-serif;
3  }
4
5  form{
6    margin: 3rem;
7    padding: 1rem;
8    border: 1px solid #ccc;
9  }
10
11 .form-control{
12   margin: 1rem 0;
13 }
14
15 label, input{
16   display: block;
17 }
18
19 label{
20   font-weight: bold;
21   margin-bottom: 0.5rem;
22 }
```